Hard Kode zine



#00
Otoño
2015

El hacklab Hard Kode comenzó como otras historias en las que ya hemos participado, fue en un departamento dentro de un viejo edificio en la Ciudad de México, en un sitio "prestado-okupado", de esta manera que el proyecto que apenas iniciaba tenia ya un futuro "incierto", al menos si hablamos del espacio físico donde se desarrollaba.

Comenzamos a organizar el hacklab, hubo varias y muy largas reuniones amenizadas por bebidas varias y con mucho punk como musica de fondo; primero queríamos tener un nombre y ponernos de acuerdo sobre lo que íbamos a hacer (y también que no queríamos hacer), se podría decir que todos los integrantes del colectivo teníamos ya algunas vivencias en organizaciones dentro del mundo hacker, por lo que al menos algo de experiencia ya había.

CRYPTO PARTY

¿QUÉ HAREMOS?
Platicaremos sobre herramientas de cifrado para correos electrónicos, mensajería instantánea, conexiones a internet, archivos y discos de tu computadora.

¿POR QUÉ LO HAREMOS?
Para experimentar juntos éstas herramientas y ejercer nuestro derecho a la privacidad

¿CUÁNDO?
Sábado 28 de marzo 2015

¿A QUÉ HORA?
16:00 HRS.

ENTRADA LIBRE

¿DONDE?
Hacklab Hard Kode: Av. Universidad 1939 departamento 301, metro Miguel Ángel de Quevedo (solo abrimos cuándo tenemos actividad programada) México, D.F.

MÁS INFO
Cryptoparty: https://www.cryptoparty.in/
HardKode: https://redmatrix.nl/channel/hardkode

HACKLAB HARD KODE

No fue fácil encontrar un nombre aunque todo apuntaba a que este tendría que ser una mezcla de punk y cultura hacker, Hard Kode fue el resultado, aludiendo a hard core dentro del punk y al código en el software libre.

Priorizamos más trabajar hacia adentro del colectivo compartiendo y socializando conocimientos, además de experimentar algunas cosas nuevas, para muchos de nosotros ya era tiempo de dejar un poco de lado los "talleres básicos" de software libre que pensamos aveces fomentan que se cree un nivel de estancamiento.

La única actividad "abierta" que se realizo en el hacklab Hard Kode fue una pequeña cryptoparty cuyo resultado fue

medianamente exitoso, l@s asistentes en su gran mayoría fueron ya gente conocida y ya con conocimientos de seguridad informática, aun así hubo un buen intercambio de conocimientos y convivencia.

Pasaron así algunos meses hasta que un día nos encontramos con la noticia de que otra gente había tomado el departamento donde nos estábamos reuniendo, si queríamos "pelear" por el espacio se necesitaría fuerza, tiempo, abogados y mucho dinero, elementos que nos costaba tener en ese momento. Y así una vez más desapareció un espacio autónomo con una temporalidad un tanto efímera, esto gracias a la especulación inmobiliaria que es un problema en México DF al igual que en otros tantos lugares del mundo.

Así que ahora el hacklab Hard Kode se mueve de manera itinerante, y estamos en busca de un espacio donde poder asentarnos de nuevo.

Este fanzine es una idea que surgió dentro del colectivo y que ahora ves plasmada entre tus manos o en la pantalla de una computadora, el Hard Kode zine no es un espacio cerrado solo para integrantes de colectivo si no que invitamos a toda la comunidad "hacker" a participar en las siguientes ediciones.

Happy hacking.

Hacklab Hard Kode

Contactanos en en Diaspora:

hardkode@pod.orkz.net

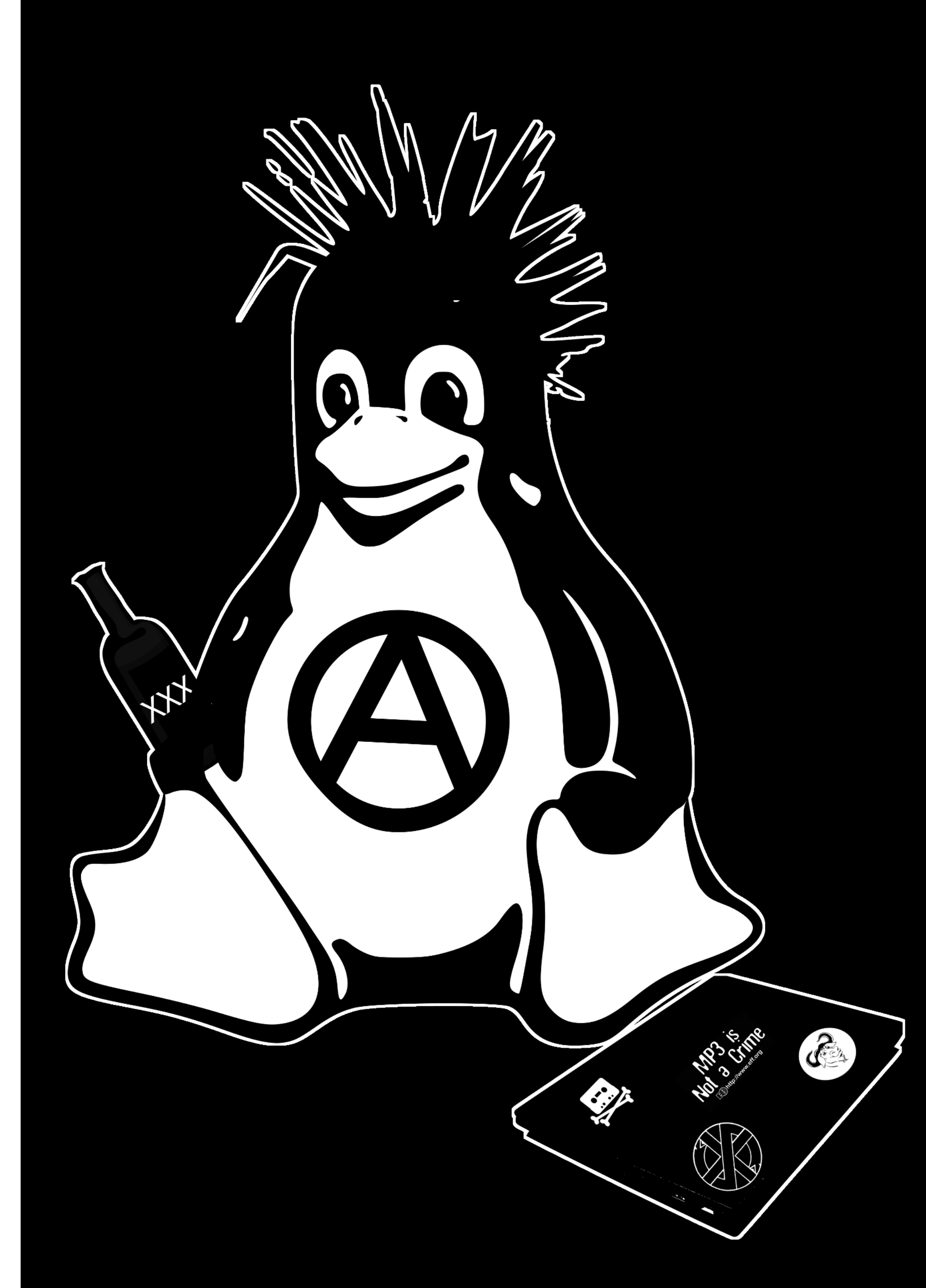
hardkode@diaspora.punkbeer.me

(Aun no decidimos completamente cual de las dos cuentas es la buena)

Y al menos de manera temporal:

www.hardkode.punksmedia.org




XXX
MP3 is
Not a Crime
http://www.eff.org

¡Al fin! Otro Hackmitin más

Anteriormente, yo ya había escrito para otras publicaciones algún resumen rápido de la historia del Hackmitin en México, aquí una más pero desde una perspectiva muy particular y nombrando algunos proyectos y sucesos que para mí son importantes para el surgimiento de éste encuentro anual.

Hace algunos años mi amigo Chiwy y yo (Pirra) estábamos adentrándonos en la cultura Hacker, primero conociendo mucho del tema gracias a los talleres y biblioweb del colectivo-servidor autónomo Espora.org del cúal después formamos parte, tiempo después empezamos a acudir a talleres, conferencias y congresos sobre temas desde GNU/Linux hasta cultura Hacker como tal, eran eventos buenos y altamente técnicos pero faltaba un toque que es lo que precisamente buscabamos y ese era trabajar con éstos conocimientos pero con un enfoque más politico, autónomo, autogestivo y con miras a la acción social. Creo que nos hacia falta eso, nosotros venimos de varios procesos desde colectivos punks, autónomos, ecologistas y zapatistas, creo que queríamos vincular todo ésto a la práctica Hacker. En medida que no encontrabamos un perfil en el cuál encajaramos al 100%, decidimos, una vez más en la vida, llevar a la práctica un principio básico que hemos tenido en varios procesos anteriores y es: "Si no te gusta, hazlo tú", una frase que mencionaba nuestro compa punk René, una frase que se escucha sencilla pero llevarla a la práctica era el reto. Entonces empezamos a organizar talleres que tenían que ver con software libre para facilitar herramientas para colectivos como para hacer páginas webs, diseños de zines y propagandas, terner algún correo electrónico en servidores autónomos, etc., algunos talleres despertaban mucho interés y algunos nada en lo absoluto. En ese tiempo acudimos a unas jornadas de talleres de software libre al sur de la ciudad de México, y fué dónde conocimos a algunos compañeros que más o menos buscaban lo mismo que nosotros, congeniamos rápidamente y nos involucramos en nuevas aventuras.

En abril del 2009 se convocaba a una edición más del FLISOL (Festival Latinoamericano de Instalación de SOftware Libre), un evento dónde se instala Software Libre a los asistentes y aparte es acompañado de charlas y presentaciones, nosotros con la idea de politizar éstas herramientas decidimos lanzar nuestro FLISOL: Abajo y a la Izqierda (Una frase


VIVE
LIBRE
O MUERE
010
001
111

reivindicativa de luchas de izquierda acuñada por el movimiento zapatista), justo en esa semana México estuvo paralizado por el brote de Influenza, una pandemia que muchos pensamos que fué una manipulación más del gobierno. La Ciudad de México se paralizó como nunca, esa semana cerraron todos los negocios y actividades, la ciudad estaba desierta y el resto de las sedes de FLISOL cambiaron de fecha a excepción del nuestro, con todo y esto fué una buena convocatoria, asistió gente y se crearon grandes lazos para crear nuevas cosas que rindieron frutos en el futuro, aunque no hicimos ni una instalación que yo recuerde.

Después de ésto, a medidados del 2009 tuvimos la oportunidad de okupar un espacio al cual le pusimos ZAM: Zona Autónoma Makhnovtchina, un espacio donde albergó actividades de diversos colectivos (punks, artistas, anarquistas, ecologistas y hasta hackers), y fué cuándo estuvimos ahí que formamos nuestro Hacklab, el Hacklab ZAM junto con la gente que estuvimos en esa ocasión en el Flisol Abajo y a la Izquierda, además de otras que se llegaron a integrar, formando un Hacklab muy diverso pero con el común denominador: muy Hacktivista. Teniamos talleres, Tech_ios, Jornadas de Software Libre, talleres de cifrado y muchas más actividades todo el tiempo, se llegó a formar una comunidad muy grande y fué cuándo tuvimos la visita de nuestro amigo Iokese de España y que nos propuso la creación de un Hackmitin y fué cuándo lo hicimos.

El hackmitin lo resumimos como un encuentro donde se reúnen hackers, promotores y usuarios de herramientas tecnológicas libres para compartir sus conocimientos. Entre esas actividades hay talleres, charlas, presentaciones y de más.
Nos gutaba mucho la idea de éste encuentro y con la influencia de los grandes hackmeetings italianos de finales de los años 80’s en espacios okupada y a partir de entonces empieza esta tradición de reunirse una vez al año en algún punto a compartir algunos días de conocimiento, desde entonces los Hackmeetings se siguen haciendo en Italia, adempás de otros paises como España, Estados Unidos, Chile, Croacia, Bolivia, México, entre otros, siempre realizandose en espacios autogestivos y autónomos y siempre tratando de romper la relación de “anfitrión-asistente” o “tallerista-alumno”.

Éstos encuentros se conocen como Hackmeetings, aunque nosotros, un poco a manera de juego y por traducirlo al español le llamamos Hackmitin (Un mitin Hacker).
El Hackmitin se sigue haciendo desde el 2009, yo he tenido la fortuna de asistir casi a todos ya que no pude participar en el del 2014, pero he visto como la comunidad ha crecido y cómo han surgido otros proyectos a partir de los Hackmitins, nosotros mismos ya hemos mutado en otros proyectos, después del desalojo de la okupa Z.A.M. se estuvo divagando un poco hasta que se creó el Hackerspace Rancho Elecrónico, gran proyecto que da continuidad a gente que estuvieron en los primeros días de la Z.A.M. y lo mejor es que se han añadido otros más en el camino, logrando ser un centro de libre compartir de conocimientos sobre herramientas tecnológicas libres, al mismo tiempo que hay otros grupos y cominidades, varios de nosotros que estuvimos o estamos en proyectos hackers y de okupaciones, ahora estamos en el Hacklab Hard Kode.

El Hackmitin ha ido más o menos asi:
-2009 “Zona Autónoma Makhnovtchina” (Z.A.M.) en México D.F.
-2010 OaHackA (Ciudad de Oaxaca) teniendo como punto de referencia una plaza pública llamada Plaza Carmen Alto ya que fueron cinco espacios autónomos como sedes: la Casa Autónoma Solidaria Oaxaqueña de Trabajo Autogestivo (C.A.S.O.T.A.), Librería Mompracem, La Jícara, Estación Cero y Barcina.
-2011 Centro de Resistencia Zapatista (CE.RE.ZA.) del municipio de Tecámac, Ojo de Agua, del Estado de México.
-2012 Acción Directa Autogestiva (A.D.A.), Ciudad de Puebla
-2013 San Cristóbal de las Casas, Chiapas en las sedes espacio cultural “El Paliacate” y “La Caverna”.
-2014 Casa del Obrero, Querétaro
-2015 San Luis Potosí.

Pirra
Diciembre 2015
http://pirra.punksmedia.org

Hardcore punk, lógica digital y cosas eléctricas

Como se menciono en el editorial de este zine, en el colectivo Hard Kode hay personas muy distintas, pero básicamente a todos nos gusta jugar/utilizar/crear/destruir con computadoras y la musica (hardcore) punk.

Como Hard Kode, dentro de la musica hardcore punk hay muchas ideas y muchas formas distintas, existen y han existido bandas involucradas en hacktivismo, movimiento maker, programación, arte electrónico, etc.

Aquí les reseñamos algunas de las bandas mas representantes.

- Destroy!

Con una canción llamada "The Revolution will not be analog" (la revolución no sera analógica) esta banda de crustcore de los años 90s, ya suena con ellos la influencia del utilizar computadoras.
2 de sus integrantes estudiaron ingeniería eléctrica y ciencias de la computación.
Nathan Smith, quien tocaba guitarra, puso en linea destroy.net, una de las primeras presencias en internet de una banda de crustpunk. allí también hay artículos acerca de "homebrewing" y al menos había un articulo científico de el: "Stack Smashing Vulnerabilities in the UNIX Operation System" (https://web.archive.org/web/19990128081520/http://destroy.net/machines/security/nate-buffer.ps)

- Creation is Crucifixion

Otra banda de mediados de los años 90s, ellos tocaban algo llamado "mathcore", se reivindicaban como anarquistas y geeks, el background de ellos mas que ingenieril era de artes. entre otros colectivos que participaban fueron Critical Ensamble (http://critical-art.net/), Carbon Defense League, y Hacktivist.net (https://web.archive.org/web/20050929021019/http://www.hacktivist.net/) entre otros. musicalmente sacaron varios discos y no de hardcore si no de algo llamado "harshnoise" (como uno en vivo en radio Bronca), tuvieron también colaboraciones con otros músicos de harshnoise siendo la mas destacable Antennacle, que en realidad era el registro de una instalación de una pieza de arte electrónico en un museo. en

todos sus discos sobran las referencias geeks y hacktivistas. uno de sus últimos eps, se llama "Child As Audience - Where Technology And Anarchy Fuck...", este es una colaboración entre varios colectivos; tales como Carbon Defense League, Critical Art Ensamble, y el Institute for Applied Autonomy. el cd incluía un sdk para compilar programas para el Z60 que contiene el asic del Gameboy, documentación, un emulador de Gameboy que se llama NO$GB, y además una rom.

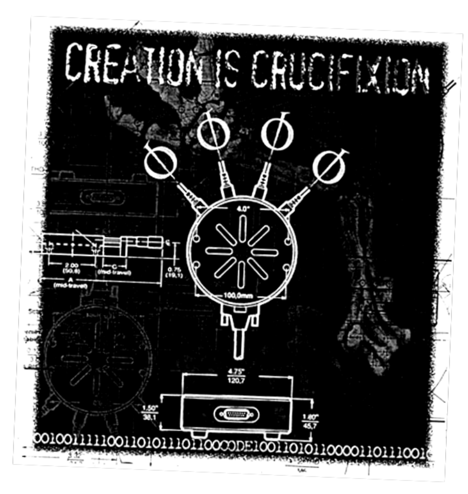
- Scholastic Deth

Ellos mas que hardcore hacktivista, son una banda de hardcore geek/nerd/etc de principios de los 2ks, sus canciones mas que referencias a computadoras, usan referencias a estudiar algo ingenieril, nombres de canciones como "Coffee Cures Everything" (el café lo cura todo) o
The Revolution Will Not Be Posted On EBay) "la revolución no sera publicada en ebay"

Destroy! y Creation Is Crucifixion son los dos ejemplos mas remarcables de hardcore punk con ondas de tecnología subversiva, no son los únicos, igual están DEV/NULL, Fate of Icarus, Axiom, entre otros, que hacen este tipo de referencias.

Para terminar, basta con decir que ni todo el hardcore punk es anti-tecnología, ni tod@s la gente del gremio son unos nerds apáticos, ni tampoco nada esta escrito sobre piedra.

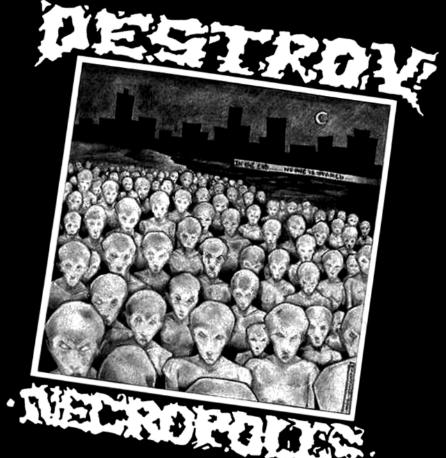
CHUKI



THE REVOLUTION WILL NOT BE ANALOG

Just as firearms
Gave the peasant a tool against the knight
So the computer is our lever against
The oppressor
Digital Revolutionaries
Wreck Havoc in cyberspace
Sabotage from within
Leaving not a trace

Our revolution will not
be analog



Products of the info-age
Raised on technology
The future is now
For the cyberpunk
Shockwave riders
At war with tyranny
Boxes, chips and fiber
optics
Stock the armory

One for the cyberpunks and phreaks out there. The system can be attacked by many means. Just as David was able to defeat Golitaht by means of a sling and superior will so too can the rebel defeat the system by superior use of its own technology. "The capitalist will gladly sell you the very rope you intend to hang him with."

Buscando libros electrónicos gratuitos en la red social libre Diaspora, me encontré con la pagina Epublibre, ahí me puse a descargar como loco todo lo que me pareció bueno y uno de los títulos que mas me intereso fue una historia de estilo policiaca pero estelarizada por hackers, el libro lleva por titulo "Takedown".

Este libro nos narra la captura de Kevin Mitnick un hacker (o quizás mejor llamarlo cracker) que disfrutaba de introducirse en servidores ajenos para hacer travesuras (típico), hasta que en una de sus intrusiones se encontró con Tsutomu Shimomura un experto en seguridad informática que al ver vulnerados sus servidores se obsesiono con no parar hasta detener a Kevin Mitnick. Hasta aquí la historia suena un poco trillada y muy posiblemente aburrida, pero lo que me gusto es que fue una historia real y nos traslada a la California en la mitad de la década de los 90 en plena efervescencia del mundo de las computadoras personales y de Internet, hay también en el libro muchas alusiones al sistema operativo Unix y sus comandos (aunque aquí se l@s traductores tuvieron el error de querer traducirlos), también se menciona a Richard Stallman y la Free Software Foundation, y algo de lo mejor es que se reivindica la imagen de hacker como alguien que tiene un espíritu creativo y rechaza la imagen que venden los medios de comunicación de hacker como "pirata informático".

En fin Takedown es una buena lectura que recomiendo.

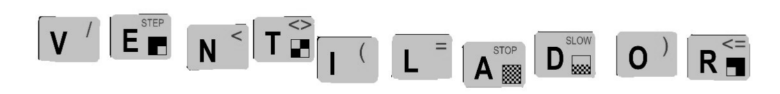
Lo puedes encontrar en:

https://www.epublibre.org/libro/detalle/4659

--
Chiwy



TU



DETECCIÓN Y CORRECCIÓN DE VULNERABILIDADES

Este texto es el primero de una serie, en la que iré explicando, como detectar vulnerabilidades en diferentes partes de un sistema y su corrección.

En esta ocasión vamos a dar una pequeña introducción y algunos términos que se utilizan constantemente, pero mas que nada abordaremos el inicio al buffer overflow.

Existen diferentes tipos de amenazas o puntos frágiles en la infinita cantidad de programas y sistemas que utilizamos día a día, estos vectores de ataque se pueden identificar no solo en los núcleos de sistemas operativos que son los mas peligrosos, si no incluso en estándares y protocolos.

Seria difícil desde este texto afirmar o aludir a que en algún sistema operativo exista mayor o menor numero de vectores de ataque, tener en cuenta que en los programas donde el código no es abierto es mas difícil encontrar este tipo de errores, pero esto no quiere decir que no existan errores a diferencia del código abierto donde se puede auditar con mayor facilidad el código y también repararlo, pero que tampoco esta exento de peligro.

Primeramente las cosas ocurren desde que se descubre una vulnerabilidad ya sea por error o por una auditoria, entonces se comienza creando un PoC (prueba de concepto), para demostrar que la vulnerabilidad existe y una posible forma de explotarla, a partir de aquí puede suceder que alguien cree un Zero-Day Exploit que en realidad ya es un código listo para ser explotado o vendido a cibercriminales para su uso, o algún gobierno lo cual no hay mucha diferencia ya que en vez de usar estos Exploits para combatir la delincuencia o esclarecer crímenes, normalmente se usan para controlar, espiar o reprimir ciudadanos inconformes a sus regímenes, de lo contrario los gobiernos harían publica la compra de este tipo de herramientas. En su mayoría los antivirus no detectarían a estos Zero-Day Exploit , otra de las cosas que pueden suceder es que algún antivirus mas experimentado gracias al PoC, agilice la prevención de vulnerabilidad creando algún sistema heurístico para su futura detección.

Un buffer overflow (o desborde de memoria) se lleva a cabo

cuando un programa informático excede el uso de cantidad de memoria asignado por el sistema operativo, escribiendo en el bloque de memoria contiguo. Estos fallos son utilizados por ciberdelincuentes para lograr ejecutar código arbitrario en un equipo, de manera que en muchos casos logran tomar control del equipo víctima o ejecutar un ataque de Denegación de Servicios (DoS).

En verdad, un buffer overflow se produce en una aplicación informática cuando no cuenta con los controles de seguridad necesarios en su código de programación. Cabe destacar que para poder llevar a cabo un desborde de memoria, se debe contar con conocimientos de programación, como también nociones básicas de arquitectura de Sistemas Operativos.

Existen diferentes formas de protección contra el desbordamiento de búfer, se utiliza para detectar los desbordamientos de búfer más comunes mediante la comprobación de que la pila no se ha alterado cuando una función la devuelve. Si se ha alterado, el programa sale con un fallo de segmentación. Tres de estos sistemas son:

1. libsafe
2. los parches gcc StackGuard
3. ProPolice.

Por ejemplo la forma de prevención de ejecución de datos de Microsoft, protege explícitamente al manejador de excepciones SEH de que se sobrescriba.

Protección de pila aun más eficiente es posible mediante el fraccionamiento de la pila en dos: una para datos y otra para los retornos de función. Esta división está presente en el lenguaje Forth, a pesar de que no fue una decisión de diseño basado en la seguridad. En cualquier caso, esto no es una solución completa a desbordamientos de búfer, ya que la dirección de retorno aún se pueden sobrescribir.

Para tener una concepción más adecuada de lo que es un buffer, es necesario recordar algunos conceptos sobre la memoria.

Estructura de la memoria

Cuando un programa es ejecutado, el sistema operativo reserva una zona de memoria para que la aplicación realice

correctamente sus instrucciones, este espacio se divide en zonas de acuerdo a 1os distintos tipos de datos. Primero es necesario cargar e1 código ejecutab1e de1 programa, es decir, 1as instrucciones. 1a zona etiquetada como data es uti1izada por 1enguajes de programación que permiten 1a creación de variab1es g1oba1es y variab1es estáticas.

Asimismo, se reservan por 1o menos dos espacios para 1os datos requeridos en 1a ejecución, estos espacios son stack y heap.

E1 stack a1macena 1os argumentos de 1as funciones, 1as variab1es 1oca1es y 1as direcciones de retorno de 1as 11amadas a funciones.

E1 heap se encarga de gestionar 1a memoria dinámica, es decir 1a memoria so1icitada durante e1 tiempo de ejecución.

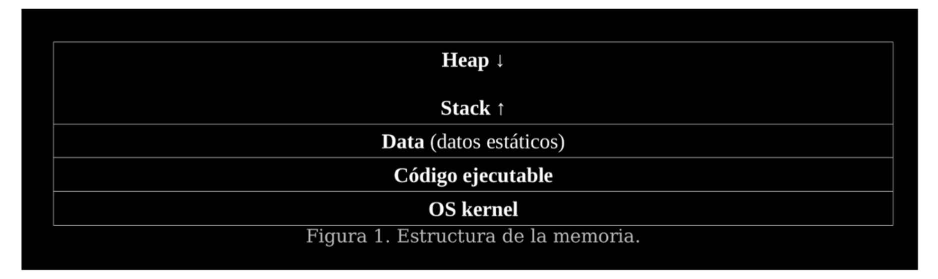


Figura 1. Estructura de la memoria.

Tipos de buffer overflow

Básicamente pueden distinguirse dos tipos primarios de buffer overflow que se ligan directamente con la explicación previa sobre las regiones de memoria ya que su nombre se deriva del espacio en memoria sobre el cual es localizada la vulnerabilidad:

- Stack overflow
- Heap overflow

En le siguiente artículo se hablará sobre el tipo stack overflow, veremos un código vu1nerab1e e1 cua1 exp1icaremos detenidamente y posteriormente procederemos a reparar e1 BUG (es un error o fa11o en un programa de computador o sistema de software que desencadena un resu1tado indeseado).

Un sa1u2 a tod@s esperando 1ean nuestro siguiente articu1o el cual sera mas practico.

Mis pininos con Retroshare

Hace ya algún tiempo que me entere de la existencia de Retroshare, un software que nos permitía comunicarnos e intercambiar archivos de forma cifrada, pero en esa ocasión me dio la impresión de que el programa estaba aun "muy verde" aun así que espere a ver si maduraba.

Y hace poco tiempo vi una noticia sobre el programa, así que por fin me decidí a probarlo.

Yo ahora estoy usando Debian Jessie y en la pagina de Retroshare hay instrucciones para añadir un repositorio a Debian y desde ahí instalar el programa, este proceso no me funciono así que descargue el paquete y lo instale mediante dpkg, esta no es mi opción favorita pero la use ahora al menos de forma temporal.

La versión de Retroshare que instale es la 0.6 y hasta ahora ha funcionado de manera perfectamente estable.

Retroshare es un software Peer to Peer (o P2P) así que no depende de ningún servidor centralizado. Cuando nos hacemos una cuenta generamos también un nuevo nodo en su red y al crear nuestra cuenta de usuario se crea también una llave GPG que es la que nos ayudara a cifrar todo, si al hacer una nueva cuenta habilitamos la opción "Opciones avanzadas" podemos tener tres opciones a la hora de crear nuestra clave, la recomendada (no se por que) de 2048bits, la High de 3072 y la mas fuerte seria la Insane de 4096 bits ¿por que no es esta la primera opción?, bueno supongo que en algunos equipos tardaría muchas horas en crearse la llave.

Ya que creamos una cuenta y entramos a Retroshare nos encontramos con un software mas complejo de lo que me gusta, por lo general prefiero los programas con pocas opciones, tal vez los que mas se apegan a la filosofía Unix "Haz una cosa y hazla bien", así que Retroshare al tener tantas opciones me da un poco de flojera para explorarlo todo, pero bueno siempre podemos empezar poco a poco.

Entre las opciones que podemos encontrar en este programa están: chat, llamadas con voz y vídeo (VoIP), correo electrónico, intercambio de archivos, foros, publicaciones donde puedes intercambiar tus enlaces a paginas favoritos,

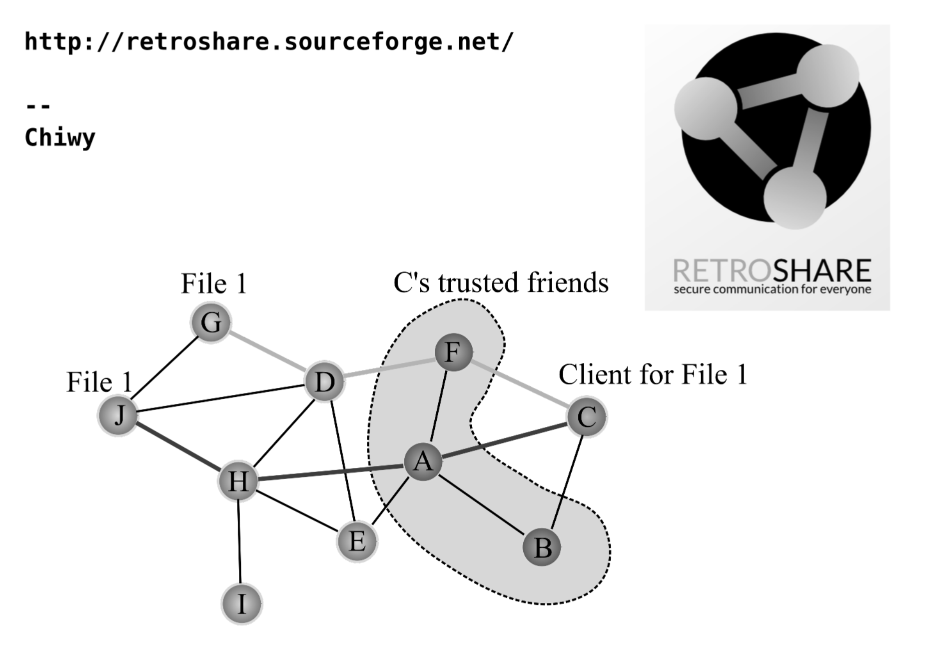
diferentes canales para publicar tus archivos, Retroshare también nos da la opción de usarlo mediante Tor.

Me llamo mucho la atención de correo electrónico sin necesidad de un servidor centralizado, pensé que así si Juanito me quiere mandar un correo yo solo tenia que estar conectado al mismo tiempo para poder recibirlo, ¿pero que pasa si por ejemplo Juanito se conecta solo por las noches y yo por las mañanas?, esta duda no me dejaba dormir, pero al indagar un poco mas la pagina del programa vi que la magia del P2P hace que si Juanito me escribe un correo y yo no estoy conectado al mismo tiempo que el los correos que me envié se pueden alojar de manera cifrada en otros nodos de amigos mutuos y me llegaran a mi cuando me conecte y alguno de estos nodos amigos estén en linea también; es simple, funciona como los torrents, mientras mas gente tenga un archivo mas fácil lo puedes obtener, así funciona también Retroshare y como cuando inicia dice algo así como "Invita amigos, Retroshare no es nada sin amigos".

Este programa es uno de mas los ejemplos de software P2P de los que cada vez hay mas, mucha gente esta enfocándose en este modelo para tener una comunicación descentralizada y fomentar así la seguridad y privacidad, en futuros números de este zine hablamos de mas ejemplos de software P2P.

http://retroshare.sourceforge.net/

--
Chiwy





WPS SIGUE EL CHIDO PA KRAKEAR REDES WPA
ATAKE PIXIE DUST ES UN ATAKE OFFLINE
KONTRA EL PIN WPS KE SAKA KLAVES EN NADA
PREUBA EL ATAKE KON PIXIESCRIPT

DESKARGA AKI:
http://sourceforge.net/projects/pixiescript/



```
if($_SERVER['HARDKODE'] == 'on')
{
```



```
}
```